BVEPreproc

Működése:

A program paramétereit a parancssorban kell átadni.
Meg kell adni az input fájl nevét, utána az eredmény fájl nevét, és a speciális jelet, amit a srciptben alkalmazni fogunk. Javasolt a & karakter.
A program futását a konzol ablakban követhetjük nyomon.

```
C:\BVETest>BVEPreproc.exe
BVE Processor (2013).
Usage:   BVEPreproc.exe <input file name> <output file name> <function character>
Example: BVEPreproc.exe in.txt out.txt "&"

C:\BVETest>BVEPreproc.exe Test2.src Test.b3d "&"
BVE Processor (2013).
Syntax error: Invalid command:'VSET'.
5:&VSET( x, 1.5 )

C:\BVETest>BVEPreproc.exe Test2.src Test.b3d "&"
BVE Processor (2013).
Finished.

C:\BVETest>_
```

A fenti példában a Test2.src fájlból elkészítjük a Teszt.b3d fájlt. De csak a harmadik nekifutásból sikerült.

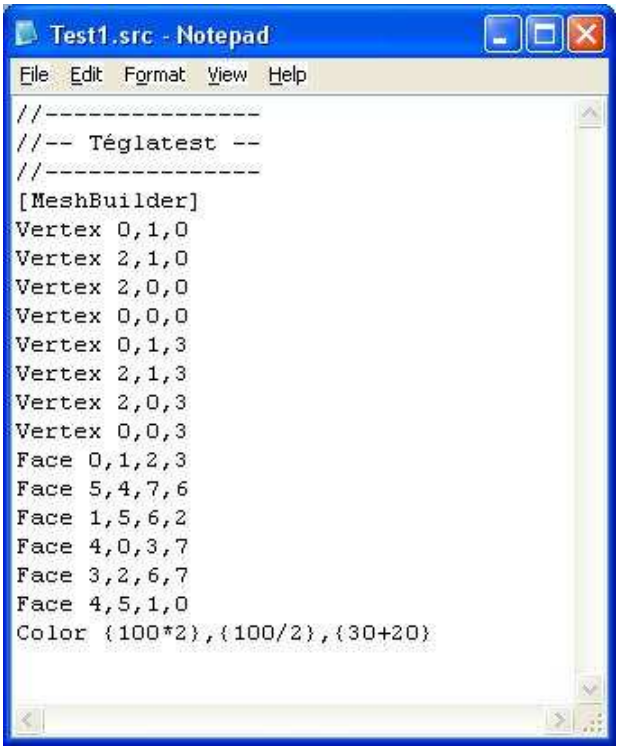
Ha a parancssor problémát okoz, készítettem mindegyik teszthez egy fordító fájlt. A végeredmény a b3d mappában keletkezik.

Teszt 1:

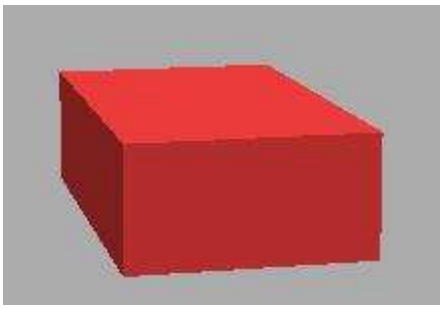
Az input fájlba megjegyzéseket írunk sor eleji // jellel kezdve.
Ezek a sorok nem kerülnek bele az eredmény fájlba.

Továbbá számolunk vele. A program a kapcsoszárójelek közé írt kifejezést kiértékeli, és az eredményt írja ki az output fájlba.

Az alábbi sorból:
`Color {100*2},{100/2},{30+20}`
Ez lesz:
`Color 200,50,50`



Az eredmény fált megnézve ez tárul elénk:



Teszt 2:

Használhatunk változókat, és ezeket a fájlba behelyettesíthetjük.
Változó létrehozása:

```
&SET( Section, [MeshBuilder] )
```

Hivatkozás:

```
&(Section)
```

Az eredmény fájlban a következő sor lesz:

```
[MeshBuilder]
```

A következő tesztfájlban a téglatest méreteit változókban tároljuk, és a Vertex parancsokban ezt behelyettesítjük.
Lássuk be, hogy ha meg szeretnénk változtatni a téglatest méreteit, az ez első három értékadással megtehetjük.
A vertexekben már az új méretek szerepelnek.

Code-Genie - [C:\BVETest\Test2.src]

File Edit Selection View Tools Window Help

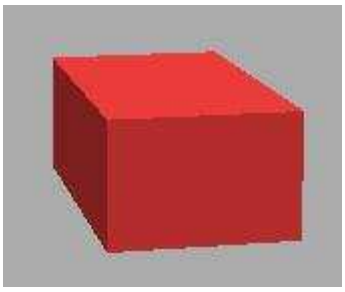
bo_dx

```
//---------------
//-- Téglatest --
//---------------
//Méretek.
&SET( x, 1.5 )
&SET( y, 1 )
&SET( z, 2.5 )
[MeshBuilder]
Vertex 0,    &(y),0
Vertex &(x),&(y),0
Vertex &(x),0,    0
Vertex 0,    0,    0
Vertex 0,    &(y),&(z)
Vertex &(x),&(y),&(z)
Vertex &(x),0,    &(z)
Vertex 0,    0,    &(z)
Face 0,1,2,3
Face 5,4,7,6
Face 1,5,6,2
Face 4,0,3,7
Face 3,2,6,7
Face 4,5,1,0
Color 200,50,50
```

Test2.src

Ln 1 Col 1 2F Loc

A végeredményen látszik, hogy kicsit változtattunk a méreteken.



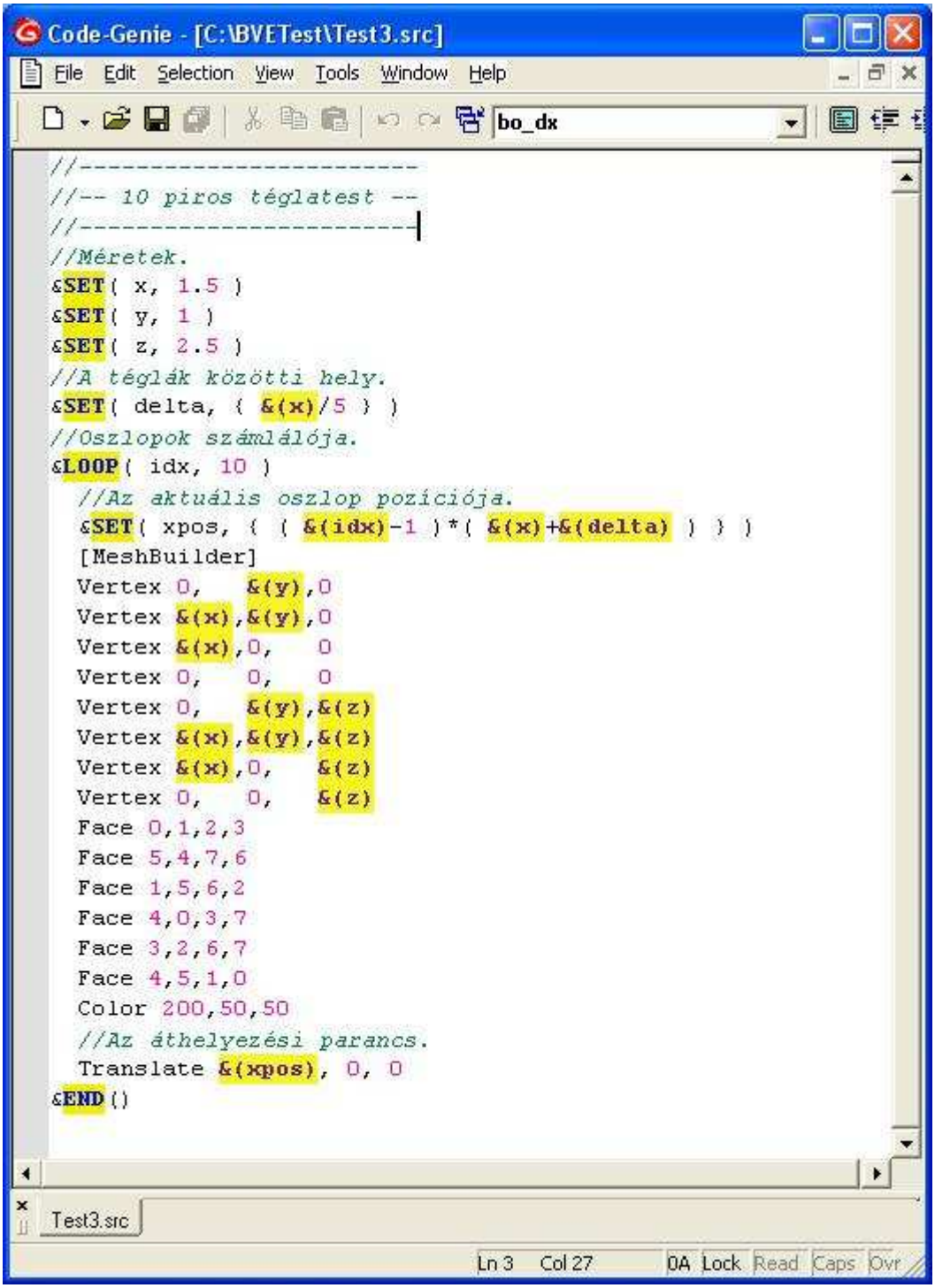
Vannak olyan szövegszerkesztő programok, ahol színezéssel látványossá thetők a forrásfájlok.
Én a Code-Genie-t használom. Lényegesen jobban néz ki, mint egy Notepad.

Teszt 3:

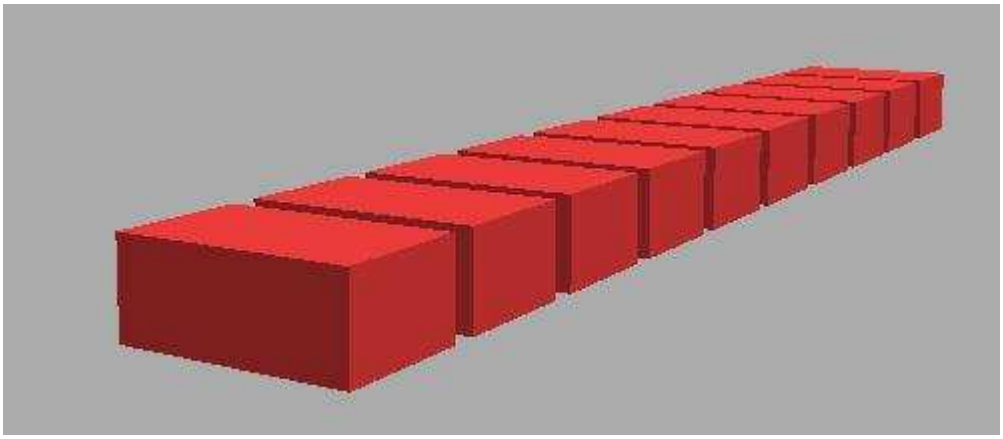
Most készítsünk 10 téglatestet, és rakjuk egymás mellé!
Definiáljuk a téglák közötti hely nagyságát. Legyen az X méret ötöde.
Figyelem: Ha számolunk, akkor kapcsos zárójelek!

```
&SET( delta, { &(x)/5 } )
```
Létrehozzuk a ciklust:
```
&LOOP( idx, 10 )
...
&END()
```
Kiszámoljuk az aktuális x pozíciót a ciklusmagban.
A ciklus számlálója a LOOP parancsban megadott idx változóban lesz.
```
&SET( xpos, { ( &(idx)-1 )*( &(x)+&(delta) ) } )
```
A végén pedig kiadjuk az áthelyezési parancsot.
```
Translate &(xpos), 0, 0
```

Figyelem: Létezik másfajta ciklus is. A LOOP az egyszerűbb.


Code-Genie - [C:\BVETest\Test3.src]
File Edit Selection View Tools Window Help
bo_dx
//------------------------
//-- 10 piros téglatest --
//------------------------
//Méretek.
&SET(x, 1.5)
&SET(y, 1)
&SET(z, 2.5)
//A téglák közötti hely.
&SET(delta, (&(x)/5))
//Oszlopok számlálója.
&LOOP(idx, 10)
//Az aktuális oszlop pozíciója.
&SET(xpos, ((&(idx)-1)*(&(x)+&(delta))))
[MeshBuilder]
Vertex 0, &(y),0
Vertex &(x),&(y),0
Vertex &(x),0, 0
Vertex 0, 0, 0
Vertex 0, &(y),&(z)
Vertex &(x),&(y),&(z)
Vertex &(x),0, &(z)
Vertex 0, 0, &(z)
Face 0,1,2,3
Face 5,4,7,6
Face 1,5,6,2
Face 4,0,3,7
Face 3,2,6,7
Face 4,5,1,0
Color 200,50,50
//Az áthelyezési parancs.
Translate &(xpos), 0, 0
&END()
Test3.src
Ln 3 Col 27
OA Lock Read Caps Ovr

Az eredmény:



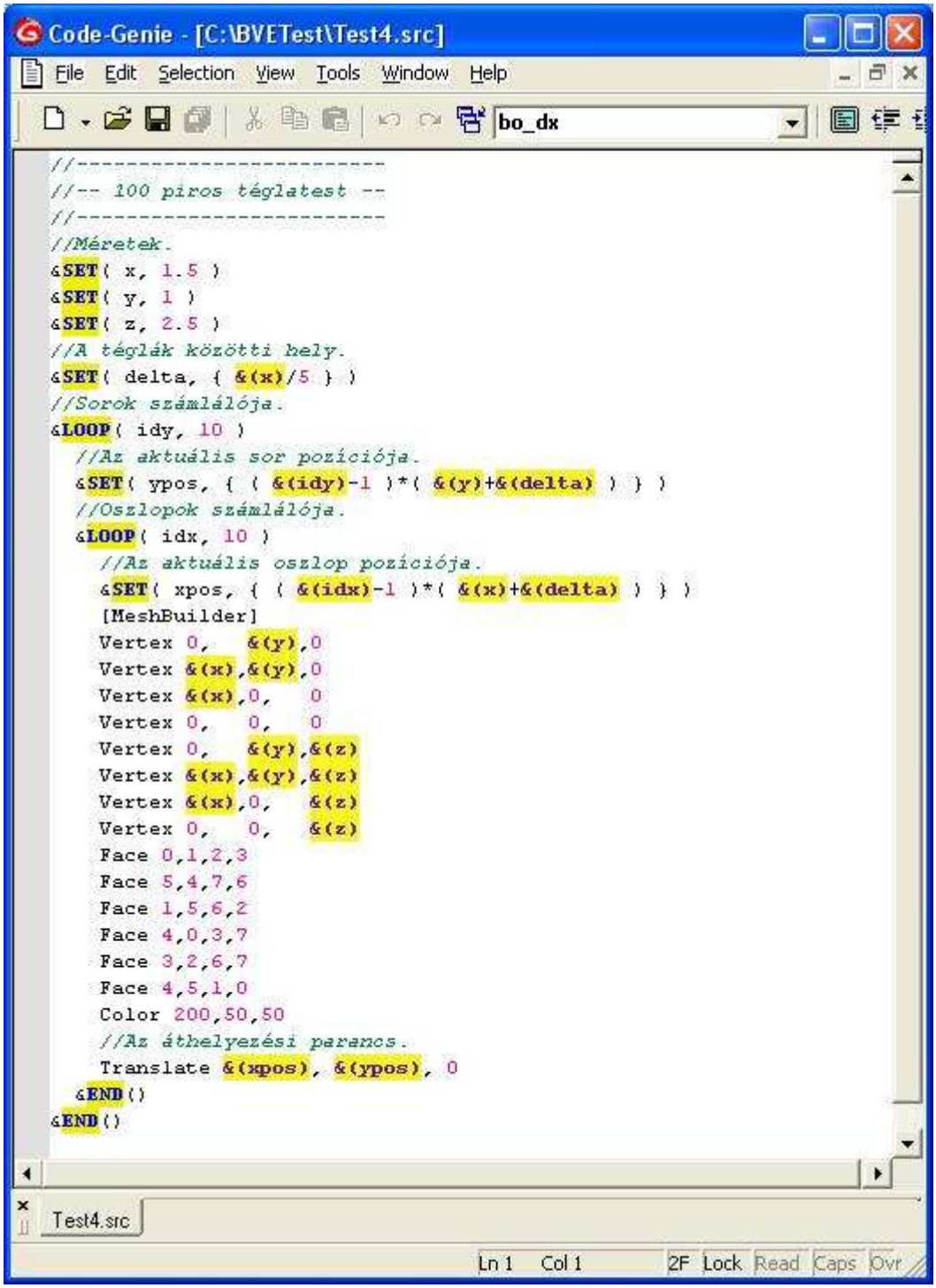
Teszt 4:

Most legyen 100 téglatest!
Az eddigi programtörzset berakjuk egy újabb ciklusba, amivel a sorokat létrehozzuk.

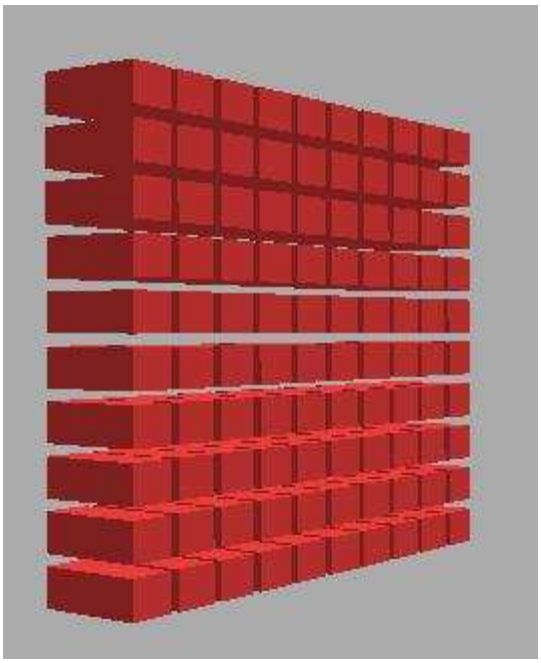
```
&LOOP( idy, 10 )
  ...
  &SET( ypos, { ( &(idy)-1 )*( &(y)+&(delta) ) } )
 ...
&END()
```

És módosítjuk az áthelyező parancsot az y koordinátával.

```
Translate &(xpos), &(ypos), 0
```



Az eredmény az, amit vártunk:



Teszt 5:

Túl egyszínűek ezek a téglák, vigyünk bele egy kis változatosságot!
A színt véletlenszerűen változtatjuk téglánként.
Kicseréljuk a Color parancsot a következőre.

```
&SET( c, { &RND(30)+20 } )
Color { &RND(50)+100+&(c) }, &(c), &(c)
```

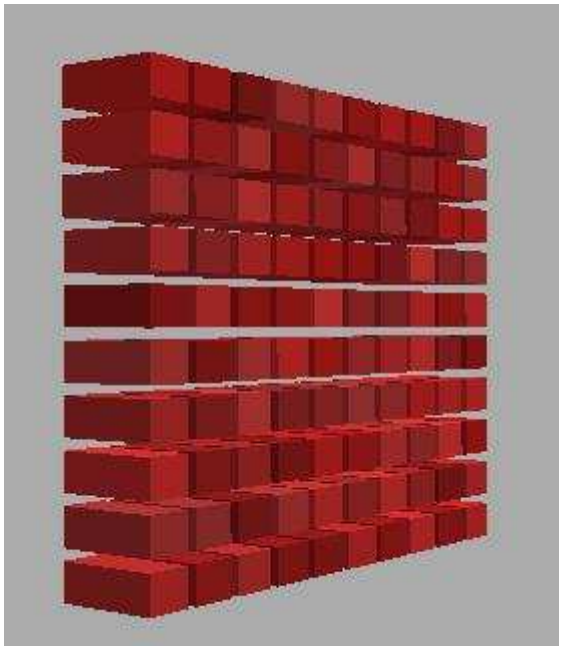
Az RND parancs egész számot generál a megadott maximális értékkel. Az

```
&RND(10)
```

parancs 1 és 10 közötti számot generál beleértve az egyet és a tízet is.

```
//-----------------------------------
//-- 100 piros árnyalatú téglatest --
//-----------------------------------
//Méretek.
&SET( x, 1.5 )
&SET( y, 1 )
&SET( z, 2.5 )
//A téglák közötti hely.
&SET( delta, { &(x)/5 } )
//Sorok számlálója.
&LOOP( idy, 10 )
  //Az aktuális sor pozíciója.
  &SET( ypos, { ( &(idy)-1 )*( &(y)+&(delta) ) } )
  //Oszlopok számlálója.
  &LOOP( idx, 10 )
    //Az aktuális oszlop pozíciója.
    &SET( xpos, { ( &(idx)-1 )*( &(x)+&(delta) ) } )
    [MeshBuilder]
    Vertex 0,   &(y),0
    Vertex &(x),&(y),0
    Vertex &(x),0,   0
    Vertex 0,   0,   0
    Vertex 0,   &(y),&(z)
    Vertex &(x),&(y),&(z)
    Vertex &(x),0,   &(z)
    Vertex 0,   0,   &(z)
    Face 0,1,2,3
    Face 5,4,7,6
    Face 1,5,6,2
    Face 4,0,3,7
    Face 3,2,6,7
    Face 4,5,1,0
    //A piros árnyalatú szín.
    &SET( c, { &RND(30)+20 } )
    Color { &RND(50)+100+&(c) }, &(c), &(c)
    //Az áthelyezési paranc.
    Translate &(xpos), &(ypos), 0
  &END()
&END()
```

Az eredmény sokkal változatosabb.



Teszt 6:

Most legyenek sárga és piros téglák vegyesen!
Az előbbi kódrészt bővítjük ilyenre.

```
&SET( c, { &RND(30)+20 } )
&IF( { &RND(2)-1 } )
  //Piros.
  Color { &RND(50)+100+&(c) }, &(c), &(c)
&ELSE()
  //Sárga.
  &SET( cc, { &RND(100)+50+&(c) } )
  Color { &(cc)+30 }, &(cc), &(c)
&END()
```

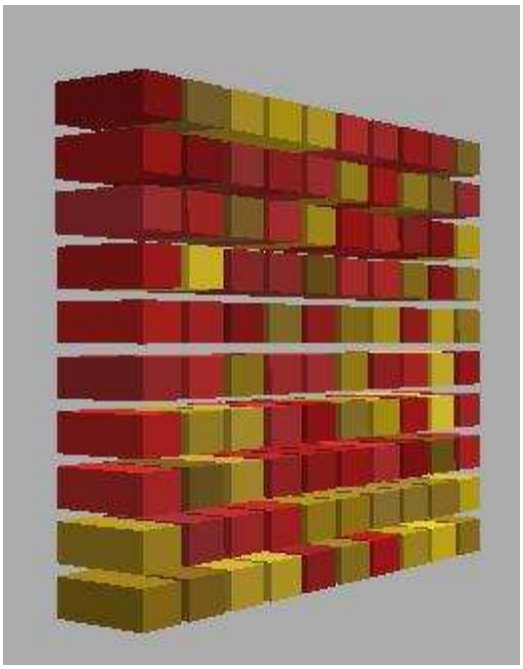
Az IF parancs argumentuma lehet nulla, vagy más.
Minden ami nem nulla az igaz.
Jelen esetben az { &RND(2)-1 } kifejezés eredménye 0 vagy 1 lehet. Ezért ugyanannyi sárga téglát várunk, mint pirosat.
Ha ezt írtuk volna { &RND(3)-1 }, akkor az eredmény 0,1,2 lehet. Mivel az 1, és a 2 érték is igaz, ezért kétszer annyi piros téglánk lenne, mint sárga.
Ez eredeti feltétel helyett írhattuk volna ezt is.

```
&IF( { &RND(2) <> 2 } )
```

Az `&RND(2)` kifejezés 1,2 értékeket ad eredményül. Piros téglát akkor rakunk ki, ha az eredmény nem egyenlő kettővel.

```
//-------------------------------------------
//-- 100 piros és sárga árnyalatú téglatest --
//-------------------------------------------
//Méretek.
#SET( x, 1.5 )
#SET( y, 1 )
#SET( z, 2.5 )
//A téglák közötti hely.
#SET( delta, ( &(x)/5 ) )
//Sorok számlálója.
#LOOP( idy, 10 )
  //Az aktuális sor pozíciója.
  #SET( ypos, ( ( &(idy)-1 )*( &(y)+&(delta) ) ) )
  //Oszlopok számlálója.
  #LOOP( idx, 10 )
    //Az aktuális oszlop pozíciója.
    #SET( xpos, ( ( &(idx)-1 )*( &(x)+&(delta) ) ) )
    [MeshBuilder]
    Vertex 0,   &(y),0
    Vertex &(x),&(y),0
    Vertex &(x),0,   0
    Vertex 0,   0,   0
    Vertex 0,   &(y),&(z)
    Vertex &(x),&(y),&(z)
    Vertex &(x),0,   &(z)
    Vertex 0,   0,   &(z)
    Face 0,1,2,3
    Face 5,4,7,6
    Face 1,5,6,2
    Face 4,0,3,7
    Face 3,2,6,7
    Face 4,5,1,0
    #SET( c, ( #RND(30)+20 ) )
    #IF( ( #RND(2)-1 ) )
      //Piros.
      Color ( #RND(50)+100+&(c) ), &(c), &(c)
    #ELSE()
      //Sárga.
      #SET( cc, ( #RND(100)+50+&(c) ) )
      Color ( &(cc)+20 ), &(cc), &(c)
    #END()
    //Az áthelyezési parancs.
    Translate &(xpos), &(ypos), 0
  #END()
#END()
```

Az eredmény



Teszt 7:

A részletes leírást mellőzve ezer téglatest kirajzolása.

```
//-------------------------------------------
//-- 1000 piros és sárga árnyalatú téglatest --
//-------------------------------------------
//Méretek.
#SET( x, 1.5 )
#SET( y, 1 )
#SET( z, 2.5 )
//A téglák közötti hely.
#SET( delta, ( &(x)/5 ) )

//Z számlálója.
#LOOP( idz, 10 )
  //Az aktuális Z pozíció.
  #SET( zpos, ( ( &(idz)-1 )*( &(z)+&(delta) ) ) )
  //Sorok számlálója.
  #LOOP( idy, 10 )
    //Az aktuális sor pozíciója.
    #SET( ypos, ( ( &(idy)-1 )*( &(y)+&(delta) ) ) )
    //Oszlopok számlálója.
    #LOOP( idx, 10 )
      //Az aktuális oszlop pozíciója.
      #SET( xpos, ( ( &(idx)-1 )*( &(x)+&(delta) ) ) )
      [MeshBuilder]
      Vertex 0,   &(y),0
      Vertex &(x),&(y),0
      Vertex &(x),0,   0
      Vertex 0,   0,   0
      Vertex 0,   &(y),&(z)
      Vertex &(x),&(y),&(z)
      Vertex &(x),0,   &(z)
      Vertex 0,   0,   &(z)
      Face 0,1,2,3
      Face 5,4,7,6
      Face 1,5,6,2
      Face 4,0,3,7
      Face 3,2,6,7
      Face 4,5,1,0
      #SET( c, ( #RND(30)+20 ) )
      #IF( ( #RND(2)-1 ) )
        //Piros.
        Color ( #RND(50)+100+&(c) ), &(c), &(c)
      #ELSE()
        //Sárga.
        #SET( cc, ( #RND(100)+50+&(c) ) )
        Color ( &(cc)+30 ), &(cc), &(c)
      #END()
      //Az áthelyezési parancs.
      Translate &(xpos), &(ypos), &(zpos)
    #END()
  #END()
#END()
```